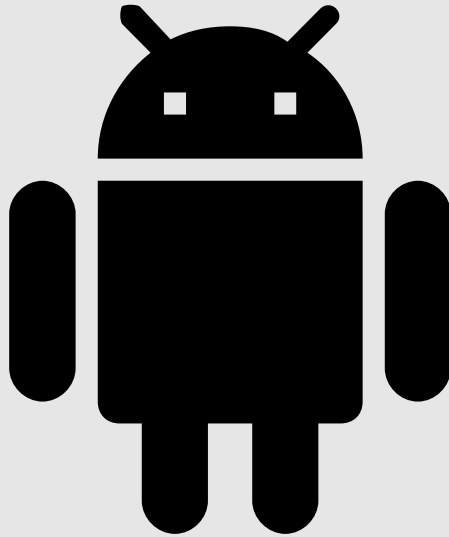


Empezar en Android con buen pie



Obture CODE

Cristina Folgueral Gutiérrez **@lady_vanilla**

Carlos Álvarez Novoa **@husky**

Consejos para hacer aplicaciones en Android

Los ejemplos son ejemplos

Separa cada aspecto de tu aplicación

¿Por qué?

Reutilización, fácil de mantener (para ti y para otros),
solidez frente a cambios

Capas

Las que necesites

Nuestras tres: Vistas, Entidades+Modelos y Datos

Datos

De donde obtenemos la información

Comunicación con web services, bases de datos o ficheros

Entidades + Modelos

Entidades: clases de negocio de tu aplicación
Modelos: usan las Entidades para responder las
necesidades del usuario

Aquí tiene que ir la lógica de tu aplicación

Vista

Parte visual

Obligatoriamente Activities y tus clases vista o Fragments

Retocar y volver a empezar

Cada proyecto es una oportunidad para utilizar y mejorar
tu arquitectura

Activity pequeña, modelo grande

Muy poco código en tu Activity

Solicitar información a los modelos y pintar
Cualquier decisión de negocio la debe tomar el modelo

Parte tu vista

Divide una pantalla en tantas vistas como necesites
Enfoca esa división a reutilización

Comunicación entre capas que permita asincronismo

¿Por qué?

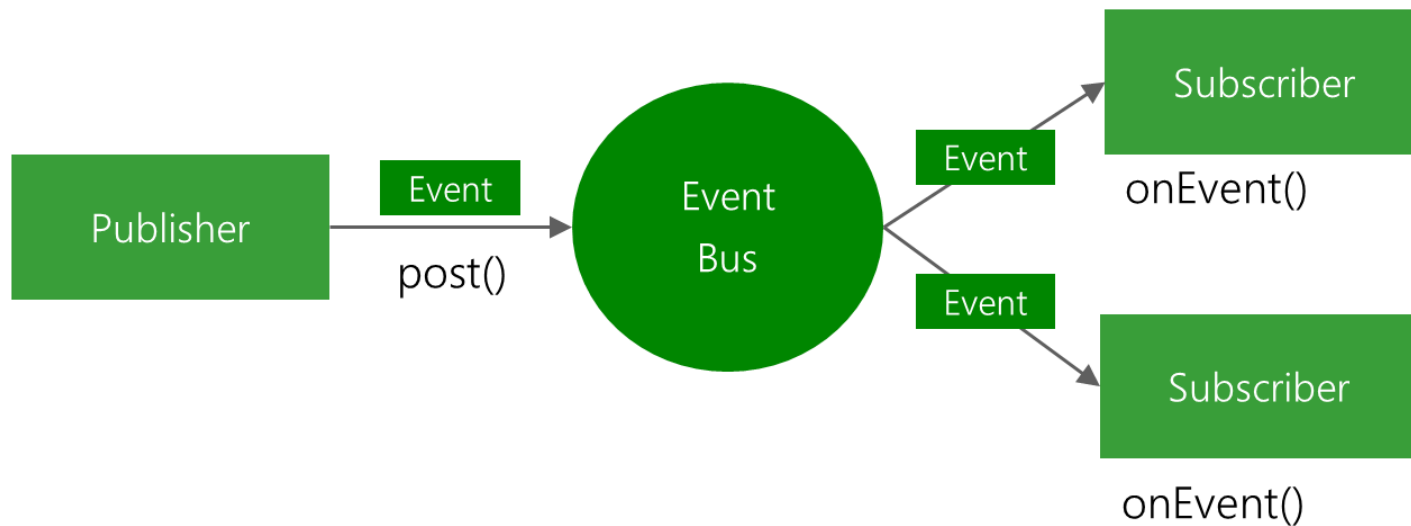
Quien pide los datos no tiene por qué saber que tiene que esperar un tiempo por ellos

¿Cómo lo podemos hacer?

Hay un montón de formas. A nosotros nos gusta utilizar el bus de eventos y/o funciones de retrollamada (patrón delegado)

Bus de eventos

Comunica tus módulos de forma eficiente manteniéndolos desacoplados



Retrollamadas

El que hace la petición recibe la respuesta en otro método definido por una interfaz.

Mantén los 60 fps

Hilo principal

Encargado de manejar eventos del sistema, refresco de pantalla, detección de gestos, etc

Cambiar vista fuera del hilo principal => Excepción

Déjalo libre

Petición a la capa de datos, parseo de información,
ordenación: en otro hilo

Opciones

AsyncTask: Cola de hilos para ejecutar tareas
Threads

Los hilos hay que pararlos manualmente

Caché, si lo tienes, no lo pidas

Ahorra datos y tiempo

Una buena política de caché permitirá ahorrar datos y mejorar la experiencia de usuario

Datos frescos VS experiencia de usuario

Factor delicado: caducidad de la caché
Dos tipos de cancelaciones: manual (por acción de usuario) y automática (al cabo de un tiempo)

Tu paquete Utils

Tu caja de herramientas

Los trozos de código que se encarguen de tareas repetitivas (formatear fechas, cadenas, etc) conviértelos en métodos estáticos
Reutilízalos en todas tus aplicaciones

Hacia dónde nos queremos dirigir

Organizar mejor la capa de vista

Implementar patrones que permitan reutilizar partes de vista y la desacoplen más del resto de capas
MVVM, MVP

Testing

Test automatizados en favor de test de aceptación
manuales

Refactoring

Introducir la refactorización como parte importante en el proceso de desarrollo de forma continua

Lee

Artículos de referencia

Clean architecture en general: <http://bit.ly/1GI1Qtw>
Clean architecture en Android: <http://bit.ly/1zSGtRB>
Patrón repositorio: <http://bit.ly/1M1HX1w>

¡Gracias!



Obture CODE

Cristina Folgueral Gutiérrez [@lady_vanilla](#)
Carlos Álvarez Novoa [@husky](#)